Avaya Solution & Interoperability Test Lab

Application Notes for Nuance Voice Platform with Avaya Aura® Session Manager and Avaya Aura® Communication Manager – Issue 1.0

Abstract

These Application Notes describe the configuration steps required to integrate the Nuance Voice Platform with Avaya Aura® Session Manager and Avaya Aura® Communication Manager using SIP integration. Nuance Voice Platform is a speech enabled telephony application that allows callers to verbally select a menu option and be transferred automatically to a third party, without waiting to speak to an operator. In addition, the callers may dial digits to select a menu option.

Information in these Application Notes has been obtained through DevConnect compliance testing and additional technical discussions. Testing was conducted via the DevConnect Program at the Avaya Solution and Interoperability Test Lab.

1. Introduction

These Application Notes describe the configuration steps required to integrate the Nuance Voice Platform (NVP) with Avaya Aura® Session Manager and Avaya Aura® Communication Manager using SIP integration. Nuance Voice Platform is a speech enabled telephony application that allows callers to verbally select a menu option and be transferred automatically to a third party, without waiting to speak to an operator. In addition, the callers may dial digits to select a menu option.

2. General Test Approach and Test Results

The interoperability compliance testing included feature and serviceability test cases.

2.1. Interoperability Compliance Testing

Feature testing focused on verifying that NVP can successfully recognize both spoken menu selections and selections entered via DTMF to transfer the call to the correct destination. Blind and supervised transfers were also verified. Other features included DNIS and CLID handling, and recording caller utterances. In addition, failover support was also tested, which verified the ability of Session Manager to re-route a call to an alternate destination if NVP was busy or unavailable.

Serviceability testing focused on verifying the ability of the NVP to recover from adverse conditions, such as server restarts, power failures, and disconnecting cables from the IP network.

2.2. Test Results

All test cases passed with the following observations noted:

- If all SIP ports on NVP are busy, the call is not re-routed by Session Manager to another destination when an alternate route is specified in its Route Policies and Dial Patterns. Session Manager does not re-route calls based on a busy condition (i.e., a "486 Busy Here" message is received). However, if a "503 Service Not Available" message is returned to Session Manager, it will re-route the call to an alternate destination, if configured properly. If NVP is down or not running, Session Manager will re-route the call.
- NVP does not support shuffling (i.e., direct IP-IP media) and should be disabled either in the IP Network Region form or SIP signaling group form on Communication Manager.
- No audio can be heard on a call after a bridge transfer is completed unless the Nuance RTP Bridging feature is enabled (**ts.RTPBridge** is set to *TRUE*).
- When accessing Nuance Management Station on a Windows 2008 x64 R2 server, there is a known issue resulting in the user being immediately redirected to the login page again when a user attempts to log in. This same behavior repeats each time the user attempts to log in. The issue can be avoided if IPv6 is disabled. To disable IPV6,

    - Use regedit, and navigate to key *[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters]*

- Right click to add a new DWORD value. Enter name as "DisabledComponents" and set its value to "FF".
- Restart the server for the change to take effect. To revert the change, simply delete the added new key and restart the server.

2.3. Support

To obtain technical support for Nuance Voice Platform, contact Nuance via email or through their website.

- **Web:** http://www.nuance.com/support/
- **Email:** support@nuance.com
- **Phone:** (866) 434-2564 or (514) 390-3922

3. Reference Configuration

Figure 1 illustrates the configuration used to verify the Nuance Voice Platform (NVP) solution with Session Manager and Communication Manager. NVP was deployed on a dedicated server running Windows Server 2008 R2. Nuance provided a customer application (*nvp4_certification.vxml*) to test the NVP solution. Session Manager interfaces to Communication Manager using a SIP trunk, and NVP interfaces to Session Manager via SIP. Calls are routed from Communication Manager to the NVP server through Session Manager. Multiple SIP ports were configured on the NVP server. System Manager was used to configure Session Manager.

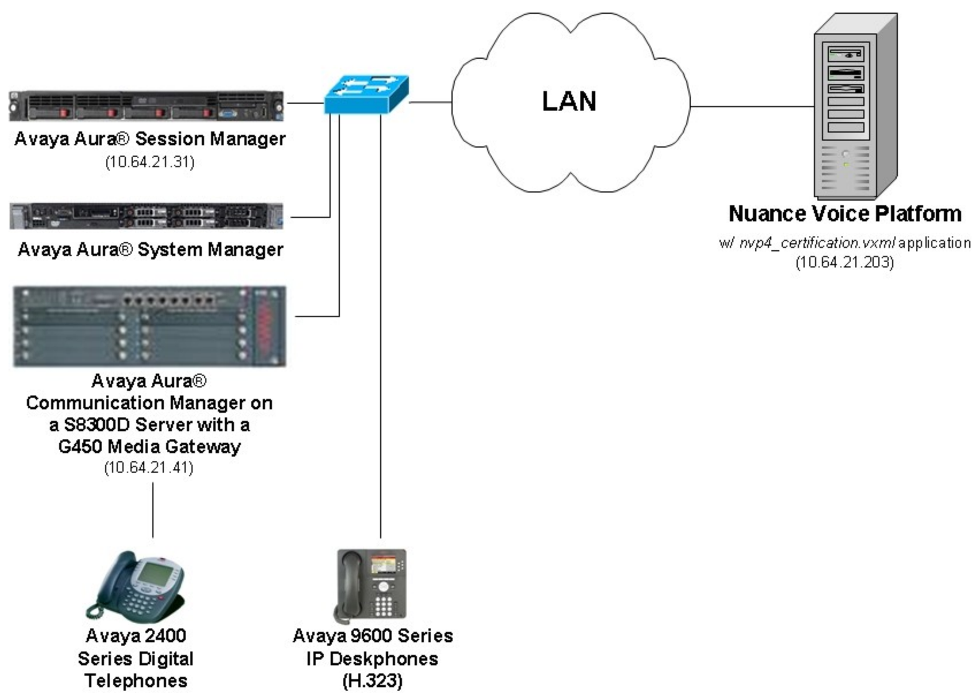


Figure 1: Configuration with Nuance Voice Platform in an Avaya SIP Network

4. Equipment and Software Validated

The following equipment and software were used for the sample configuration:

Equipment	Software
Avaya S8300D Server with a G450 Media Gateway	Avaya Aura® Communication Manager 5.2.1 Update 19196 (combo SP9 + memleakfix)
Avaya Aura® Session Manager	6.1. (6.1.3.0.613006)
Avaya Aura® System Manager	6.1.0 (Build No. - 6.1.0.0.7345-6.1.5.106), Software Update Revision No : 6.1.6.1.1087
Avaya 9600 Series IP Telephones	3.110b (H.323)
Nuance Voice Platform (NVP)	4.0.2 (Service Pack 2) with Patch NVP4.0.2_CS_for_NNH_patch_20110708_CT75807-Windows

5. Configure Avaya Aura® Communication Manager

This section provides the procedures for configuring Communication Manager. The procedures include the following areas:

- Verify Communication Manager license
- Configure a SIP trunk between Communication Manager and Session Manager
- Configure AAR Call Routing to NVP

Use the System Access Terminal (SAT) to configure Communication Manager and log in with the appropriate credentials.

5.1. Verify SIP Trunk Capacity

Using the SAT, verify that SIP trunks are enabled in the **system-parameters customer-options** form. The license file installed on the system controls these options. If a required feature is not enabled, contact an authorized Avaya sales representative.

On **Page 2** of the **system-parameters customer-options** form, verify that the number of SIP trunks supported by the system is sufficient.

```
display system-parameters customer-options                      Page   2 of  11
                                OPTIONAL FEATURES

IP PORT CAPACITIES                                            USED
                     Maximum Administered H.323 Trunks: 450   83
           Maximum Concurrently Registered IP Stations: 450   2
             Maximum Administered Remote Office Trunks: 0     0
Maximum Concurrently Registered Remote Office Stations: 0     0
              Maximum Concurrently Registered IP eCons: 0     0
  Max Concur Registered Unauthenticated H.323 Stations: 50    0
                        Maximum Video Capable Stations: 50    0
                   Maximum Video Capable IP Softphones: 50    0
                       Maximum Administered SIP Trunks: 450   94
  Maximum Administered Ad-hoc Video Conferencing Ports: 0     0
   Maximum Number of DS1 Boards with Echo Cancellation: 0     0
                             Maximum TN2501 VAL Boards: 0     0
                     Maximum Media Gateway VAL Sources: 20    1
           Maximum TN2602 Boards with 80 VoIP Channels: 0     0
          Maximum TN2602 Boards with 320 VoIP Channels: 0     0
   Maximum Number of Expanded Meet-me Conference Ports: 0     0

         (NOTE: You must logoff & login to effect the permission changes.)
```

5.2. Configure SIP Trunk

In the **IP Node Names** form, assign an IP address and host name for the Session Manager SIP interface. The processor (i.e. **procr**) and Session manager (i.e. **SM_21_31**) host names will be used for other configuration screens within Communication Manager.

```
change node-names ip                                              Page   1 of   2
                                  IP NODE NAMES
    Name              IP Address
S8500FAX            10.64.22.16
SES                 10.64.21.112
SM                  10.64.20.31
SM_21_31            10.64.21.31
SagemInterstar      10.64.22.160
default             0.0.0.0
msgserver           10.64.20.63
procr               10.64.21.111
```

In the **IP Network Region** form, the **Authoritative Domain** field is configured to match the domain name configured on Session Manager. In this configuration, the domain name is *avaya.com*. By default, **IP-IP Direct Audio** (shuffling) is enabled to allow audio traffic to be sent directly between IP endpoints without using media resources in the Avaya G450 Media Gateway. However, NVP does not support shuffling, so it should be disabled either in the IP Network Region form or in the SIP signaling group form. The **IP Network Region** form also specifies the **IP Codec Set** to be used for calls routed over the SIP trunk to Session Manager. This codec set is used when its corresponding network region (i.e., IP Network Region ‘1’) is specified in the SIP signaling group.

```
change ip-network-region 1                                        Page   1 of  19
                               IP NETWORK REGION
  Region: 1
Location: 1       Authoritative Domain: avaya.com
    Name:
MEDIA PARAMETERS              Intra-region IP-IP Direct Audio: no
      Codec Set: 1            Inter-region IP-IP Direct Audio: no
   UDP Port Min: 2048                    IP Audio Hairpinning? y
   UDP Port Max: 35889
DIFFSERV/TOS PARAMETERS                RTCP Reporting Enabled? y
 Call Control PHB Value: 48      RTCP MONITOR SERVER PARAMETERS
        Audio PHB Value: 48       Use Default Server Parameters? y
        Video PHB Value: 26
802.1P/Q PARAMETERS
 Call Control 802.1p Priority: 6
        Audio 802.1p Priority: 6
        Video 802.1p Priority: 5      AUDIO RESOURCE RESERVATION PARAMETERS
H.323 IP ENDPOINTS                                   RSVP Enabled? n
  H.323 Link Bounce Recovery? y
 Idle Traffic Interval (sec): 20
   Keep-Alive Interval (sec): 5
            Keep-Alive Count: 5
```

In the **IP Codec Set** form, select the audio codec type supported for calls routed over the SIP trunk to NVP. The form is accessed via the **change ip-codec-set 1** command. Note that IP codec set '1' was specified in IP Network Region '1' shown above. The default settings of the **IP Codec Set** form are shown below. NVP supports G.711.

```
change ip-codec-set 1                                          Page   1 of   2

                          IP Codec Set

    Codec Set: 1

    Audio        Silence      Frames   Packet
    Codec        Suppression  Per Pkt  Size(ms)
 1: G.711MU          n          2        20
 2:
 3:
 4:
 5:
 6:
 7:


     Media Encryption
 1: none
 2:
 3:
```

Prior to configuring a SIP trunk group for communication with Session Manager, a SIP signaling group must be configured. Configure the Signaling Group form as follows:

- Set the **Group Type** field to *sip*.
- The **Transport Method** field was set to *tls*.
- Set the **IMS Enabled** field to *n*.
- Specify the processor and Session Manager as the two ends of the signaling group in the **Near-end Node Name** field and the **Far-end Node Name** field, respectively. These field values are taken from the **IP Node Names** form.
- Ensure that the TLS port value of *5061* is configured in the **Near-end Listen Port** and the **Far-end Listen Port** fields.
- The preferred codec for the call will be selected from the IP codec set assigned to the IP network region specified in the **Far-end Network Region** field.
- Enter the domain name of Session Manager in the **Far-end Domain** field. In this configuration, the domain name is *avaya.com*.
- The **DTMF over IP** field should be set to the default value of *rtp-payload*. Communication Manager supports DTMF transmission using RFC 2833.
- The **Direct IP-IP Audio Connections** field was disabled on this form since NVP does not support shuffling.

The default values for the other fields may be used.

```
add signaling-group 1                                              Page   1 of   1
                                SIGNALING GROUP

 Group Number: 1              Group Type: sip
                        Transport Method: tls
  IMS Enabled? n
     IP Video? n




   Near-end Node Name: procr                  Far-end Node Name: SM_21_31
 Near-end Listen Port: 5061                 Far-end Listen Port: 5061
                                         Far-end Network Region: 1
Far-end Domain: avaya.com

                                          Bypass If IP Threshold Exceeded? n
Incoming Dialog Loopbacks: eliminate               RFC 3389 Comfort Noise? n
         DTMF over IP: rtp-payload         Direct IP-IP Audio Connections? n
Session Establishment Timer(min): 3                  IP Audio Hairpinning? n
         Enable Layer 3 Test? n                    Direct IP-IP Early Media? n
H.323 Station Outgoing Direct Media? n           Alternate Route Timer(sec): 6
```

Configure the **Trunk Group** form as shown below. This trunk group is used for calls to SIP endpoints. Set the **Group Type** field to *sip*, set the **Service Type** field to *tie*, specify the signaling group associated with this trunk group in the **Signaling Group** field, and specify the **Number of Members** supported by this SIP trunk group. Configure the other fields in bold and accept the default values for the remaining fields.

```
add trunk-group 1                                              Page   1 of  21
                                TRUNK GROUP

Group Number: 1                    Group Type: sip          CDR Reports: y
  Group Name: to SM_21_31                COR: 1        TN: 1        TAC: *001
   Direction: two-way       Outgoing Display? n
 Dial Access? n                                   Night Service:
Queue Length: 0
Service Type: tie                  Auth Code? n

                                                       Signaling Group: 1
                                                     Number of Members: 10
```

On **Page 3** of the trunk group form, set the **Numbering Format** field to *private*. This field specifies the format of the calling party number sent to the far-end.

```
add trunk-group 1                                              Page   3 of  21
TRUNK FEATURES
          ACA Assignment? n            Measured: none
                                                          Maintenance Tests? y



                     Numbering Format: private
                                                UUI Treatment: service-provider

                                                 Replace Restricted Numbers? n
                                                Replace Unavailable Numbers? n







 Show ANSWERED BY on Display? y
```

Configure the **Private Numbering Format** form to send the calling party number to the far-end. Add an entry so that local stations with a 4-digit extension beginning with '3' whose calls are routed over SIP trunk group "1", have the extension sent to the NVP for proper CLID handling.

```
change private-numbering 0                                     Page   1 of   2
                           NUMBERING - PRIVATE FORMAT

Ext Ext              Trk        Private           Total
Len Code             Grp(s)     Prefix            Len
 4  3                1                            4     Total Administered: 5
                                                           Maximum Entries: 540
```

5.3. Configure AAR Call Routing to Nuance Voice Platform

In the **AAR Digit Analysis Table**, specify a **Dialed String** that would match the NVP extension (2222) and specify the **Route Pattern** that will be used to route the call.

```
change aar analysis 2                                           Page   1 of   2
                             AAR DIGIT ANALYSIS TABLE
                                   Location:  all          Percent Full:    0

          Dialed           Total      Route     Call   Node  ANI
          String         Min  Max  Pattern    Type   Num   Reqd
        2222                4    4      1        aar          n
```

In the *Route Pattern* form, specify the SIP trunk group and set the **Numbering Format** for the route preference to *unk-unk*. This would prevent a + sign to be prepended to the dialed digits in conjunction with using private numbering format in the SIP trunk group.

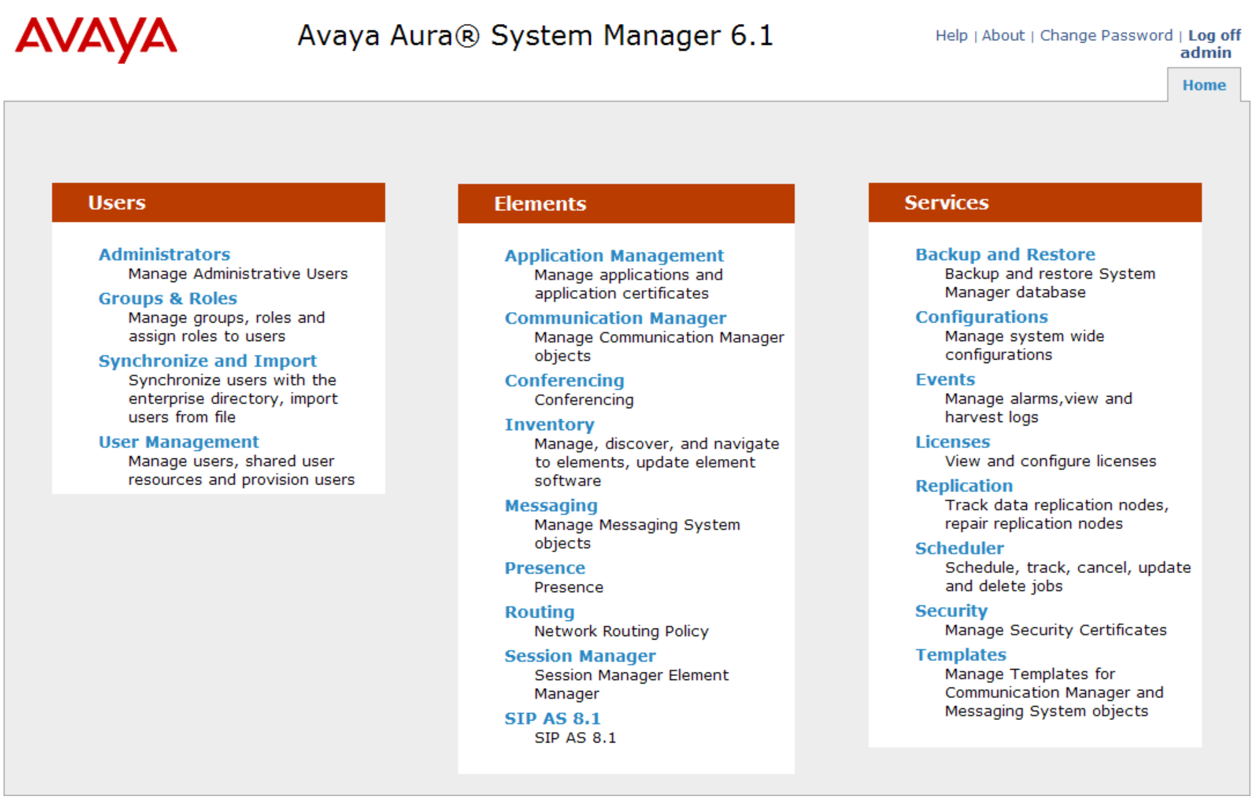
```
change route-pattern 1                                          Page   1 of   3
                    Pattern Number: 1   Pattern Name: to SM_21_31
                             SCCAN? n     Secure SIP? n
    Grp FRL NPA Pfx Hop Toll No.  Inserted                              DCS/ IXC
    No          Mrk Lmt List Del  Digits                                QSIG
                             Dgts                                       Intw
 1: 1    0                    0                                          n   user
 2:                                                                      n   user
 3:                                                                      n   user
 4:                                                                      n   user
 5:                                                                      n   user
 6:                                                                      n   user

     BCC VALUE  TSC CA-TSC    ITC BCIE Service/Feature PARM  No. Numbering LAR
    0 1 2 M 4 W     Request                                 Dgts Format
                                                         Subaddress
 1: y y y y y n  n            rest                                unk-unk   none
 2: y y y y y n  n            rest                                          none
 3: y y y y y n  n            rest                                          none
 4: y y y y y n  n            rest                                          none
 5: y y y y y n  n            rest                                          none
```

6. Configure Avaya Aura® Session Manager

This section provides the procedures for configuring Session Manager. The procedures include adding the following items:

- SIP domain
- Logical/physical Locations that can be occupied by SIP Entities
- SIP Entities corresponding to Session Manager and Communication Manager
- Entity Links, which define the SIP trunk parameters used by Session Manager when routing calls to/from SIP Entities
- Routing Policies, which control call routing between the SIP Entities
- Dial Patterns, which govern to which SIP Entity a call is routed
- Session Manager, corresponding to the Session Manager Server to be managed by System Manager

Configuration is accomplished by accessing the browser-based GUI of System Manager using the URL "https://<*ip-address*>/SMGR", where <*ip-address*> is the IP address of System Manager. After logging in with the appropriate credentials, the **Home** screen shown below is displayed.



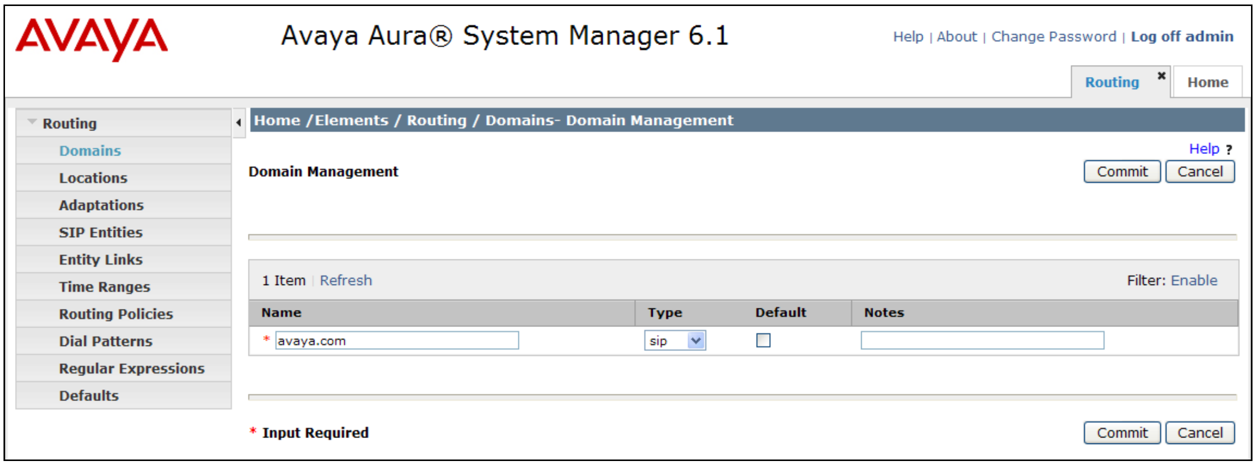
6.1. Specify SIP Domain

Add the SIP domain for which the communications infrastructure will be authoritative. Starting from the Home screen, navigate to **Routing → Domains** and click the **New** button (not shown) on the right. The following screen will then be displayed. Fill in the following:

- **Name:** The authoritative domain name (e.g., *avaya.com*)
- **Type:** *sip*
- **Notes:** Descriptive text (optional).

Click **Commit**.

Since the sample configuration does not deal with any other domains, no additional domains need to be added.



6.2. Add Locations

Locations can be used to identify logical and/or physical locations where SIP Entities reside for purposes of bandwidth management. To add a Location, navigate to **Routing → Locations** from the Home screen and click on **New** button (not shown) on the right. The following screen will then be displayed. Fill in the following:

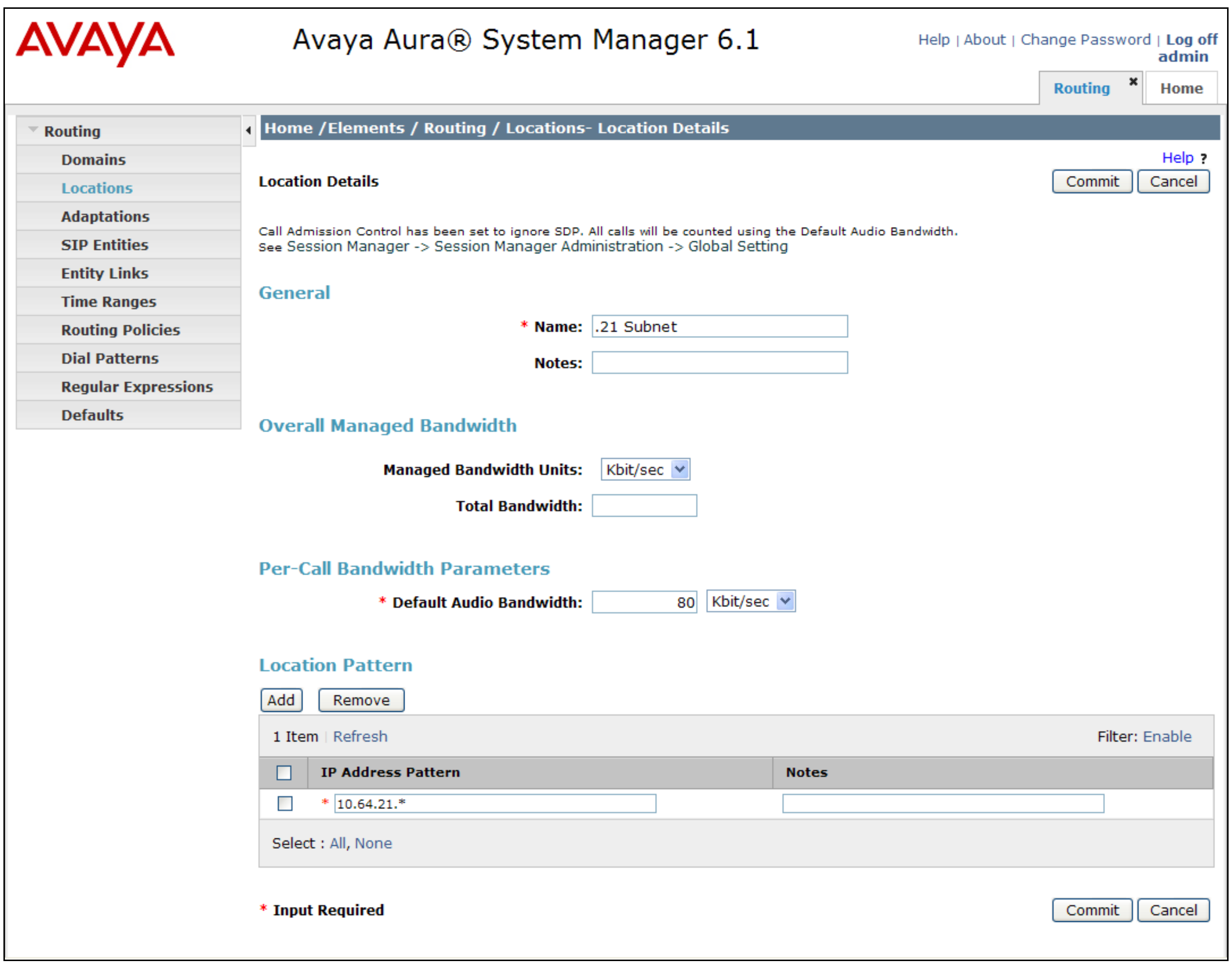
Under *General*:

- **Name:** A descriptive name.
- **Notes:** Descriptive text (optional).

Under *Location Pattern*:

- **IP Address Pattern:** A pattern used to logically identify the location.
- **Notes:** Descriptive text (optional).

The screen below shows addition of the *.21 Subnet* location, which includes the Communication Manager and Session Manager. Click **Commit** to save the Location definition.



6.3. Add SIP Entities

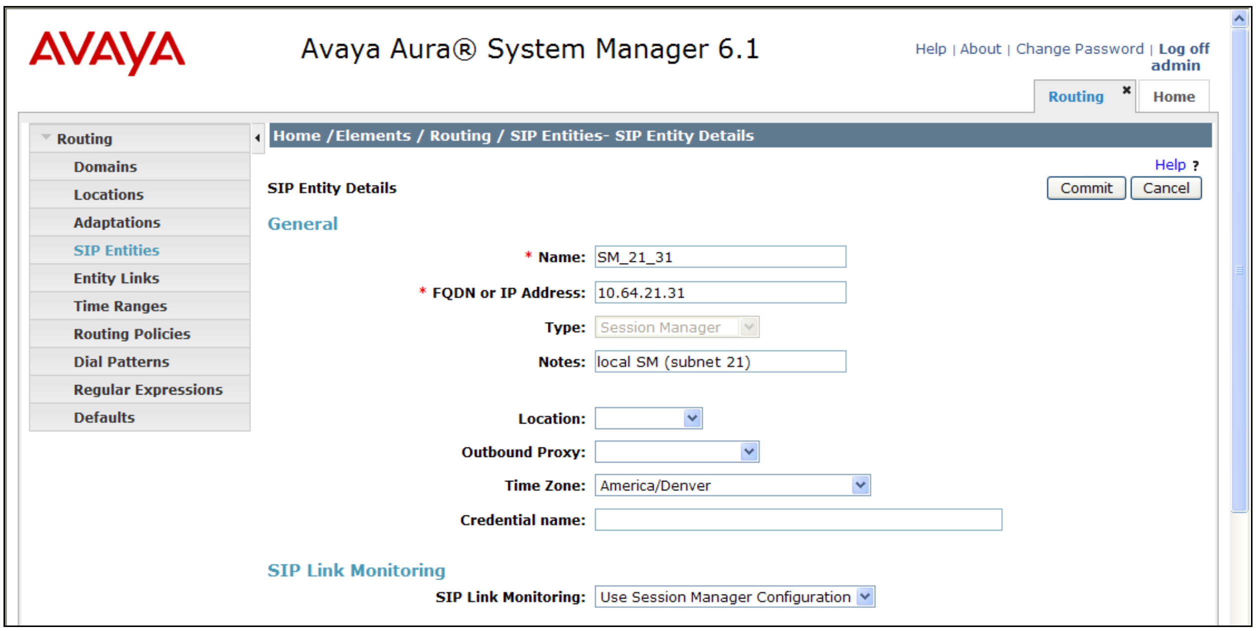
In the sample configuration, a SIP Entity is added for Session Manager, Communication Manager, and Nuance Voice Platform.

6.3.1. Avaya Aura® Session Manager

A SIP Entity must be added for Session Manager. To add a SIP Entity, navigate to **Routing → SIP Entities** from the Home screen and click the **New** button (not shown) on the right. The following screen is displayed. Fill in the following:

Under *General*:

- **Name:** A descriptive name.
- **FQDN or IP Address:** IP address of the signaling interface on Session Manager.
- **Type:** Select *Session Manager*.
- **Location:** Select the location defined previously.
- **Time Zone:** Time zone for this location.



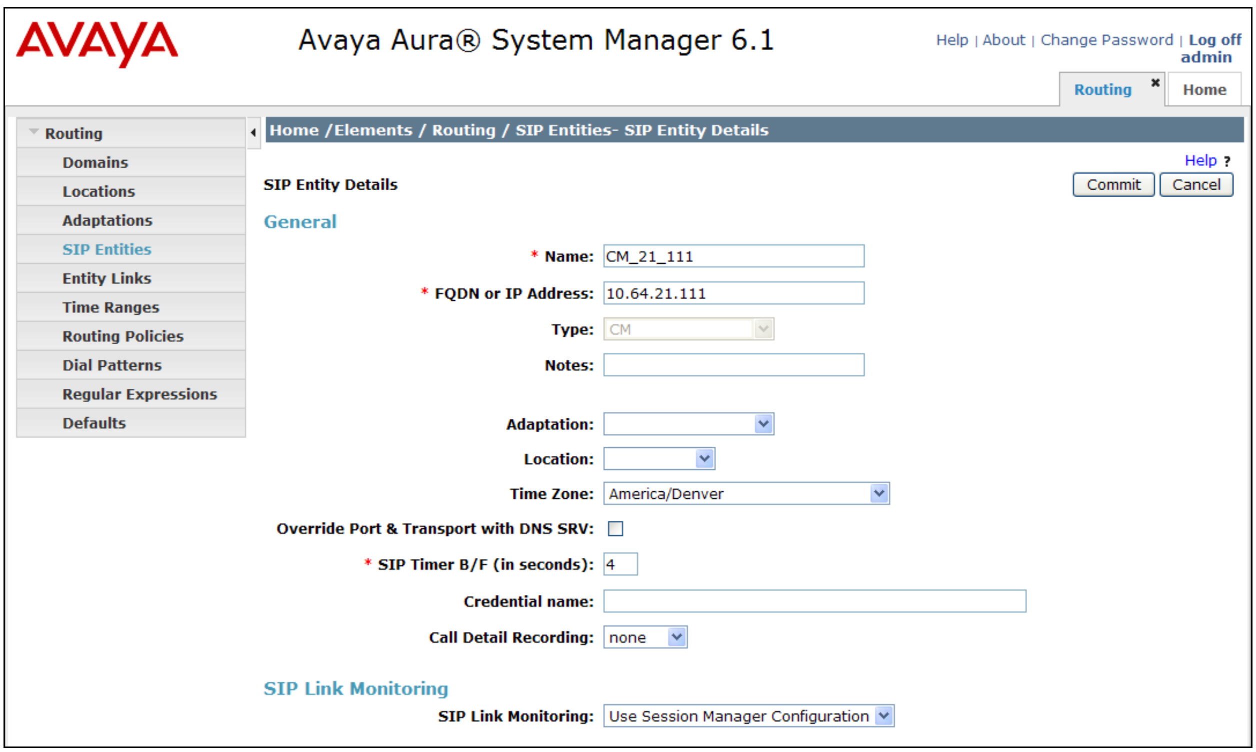
6.3.2. Avaya Aura® Communication Manager

A SIP Entity must be added for the Communication Manager. To add a SIP Entity, navigate to **Routing → SIP Entities** from the Home screen and click on the **New** button (not shown) on the right. The following screen is displayed. Fill in the following:

Under *General*:

- **Name:** A descriptive name.
- **FQDN or IP Address:** IP address of the signaling interface (e.g., procr) on the telephony system.
- **Type:** Select *CM*.
- **Location:** Select the location defined previously.
- **Time Zone:** Time zone for this location.

Defaults can be used for the remaining fields. Click **Commit** to save each SIP Entity definition.



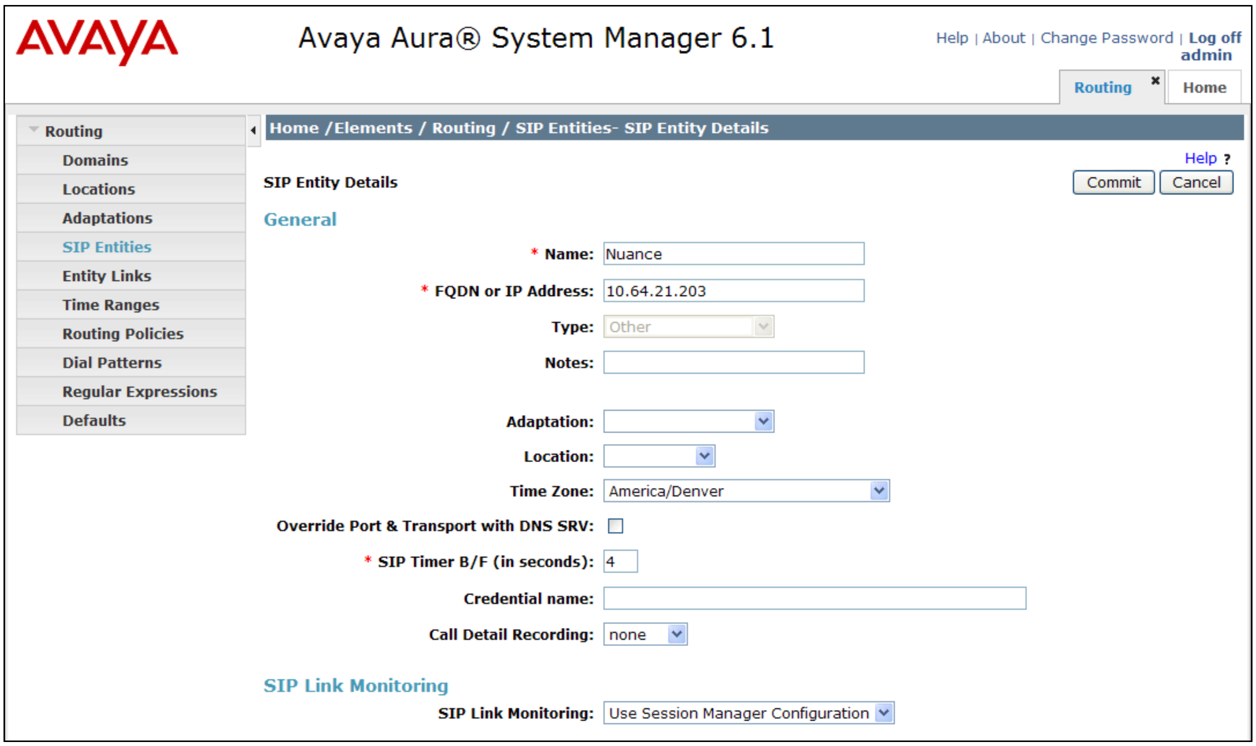
6.3.3. Nuance Voice Platform

A SIP Entity must be added for NVP. To add a SIP Entity, navigate to **Routing → SIP Entities** from the Home screen and click the **New** button (not shown) on the right. The following screen is displayed. Fill in the following:

Under *General*:

- **Name:** A descriptive name.
- **FQDN or IP Address:** NVP IP address.
- **Type:** Select *Other*.
- **Time Zone:** Time zone for this location.

Defaults may be used for the remaining fields. Click **Commit** to save each SIP Entity definition.



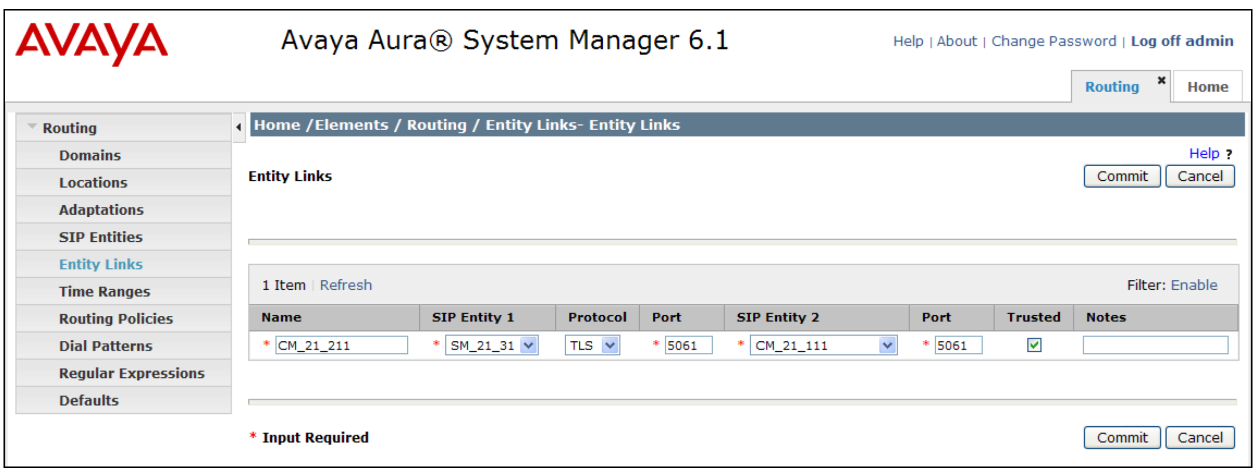
6.4. Add Entity Link

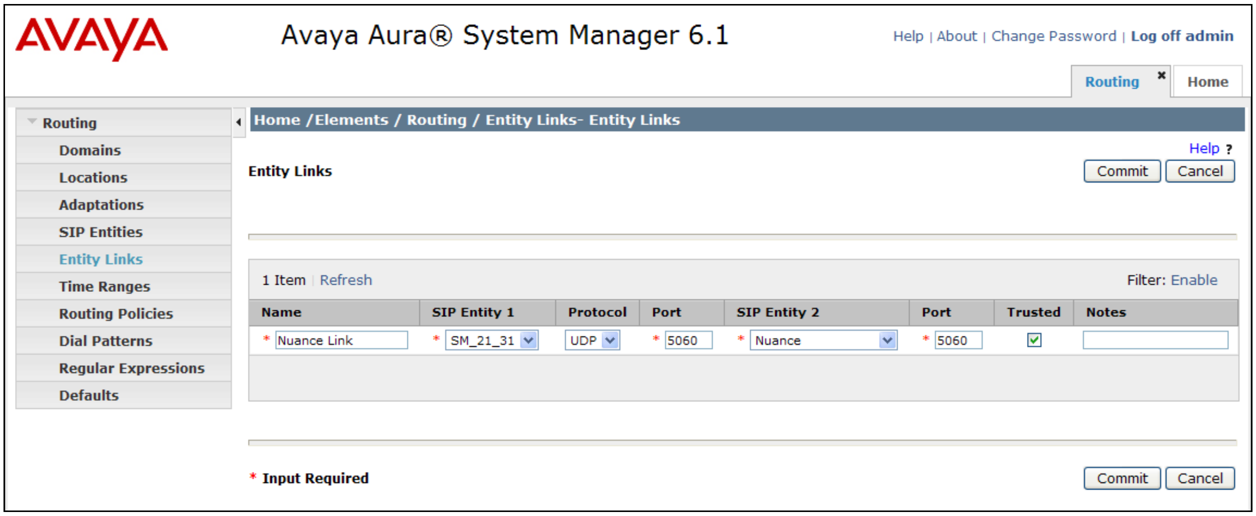
The SIP trunk from Session Manager to Communication Manager and NVP are described by Entity Links. To add an Entity Link, navigate to **Routing → Entity Links** from the Home screen and click on the **New** button (not shown) on the right. Fill in the following fields in the new row that is displayed:

- **Name:** A descriptive name.
- **SIP Entity 1:** Select the Session Manager.
- **Protocol:** Select the appropriate protocol.
- **Port:** Port number to which the other system sends SIP requests.
- **SIP Entity 2:** Select the name of Communication Manager or NVP.
- **Port:** Port number on which the other system receives SIP requests.
- **Trusted:** Check this box.

.

The following screens display the configuration of the two Entity Links per the instructions above. The first entity link is for Session Manager and Communication Manager and the second entity link is for Session Manager and NVP.




AVAYA
Avaya Aura® System Manager 6.1
Help | About | Change Password | Log off admin
Routing
Home
Routing
Domains
Locations
Adaptations
SIP Entities
Entity Links
Time Ranges
Routing Policies
Dial Patterns
Regular Expressions
Defaults
Home /Elements / Routing / Entity Links- Entity Links
Help ?
Entity Links
Commit Cancel
1 Item | Refresh
Filter: Enable
Name SIP Entity 1 Protocol Port SIP Entity 2 Port Trusted Notes
* Nuance Link * SM_21_31 UDP * 5060 * Nuance * 5060
* Input Required
Commit Cancel

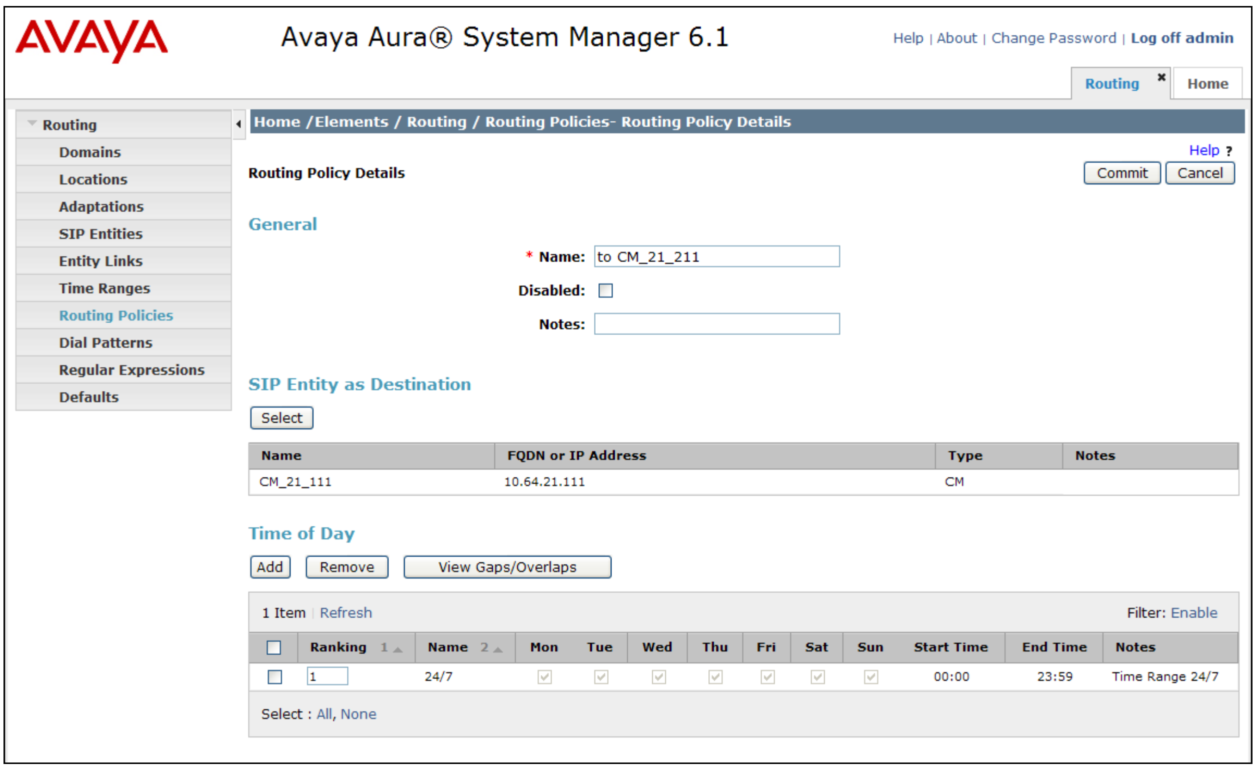
6.5. Add Routing Policies

Routing policies describe the conditions under which calls will be routed to the SIP Entities. Two routing policies were added – one for Communication Manager and one for NVP. To add a Routing Policy, navigate to **Routing → Routing Policies** from the Home screen and click the **New** button (not shown) on the right. The following screen is displayed. Fill in the following:

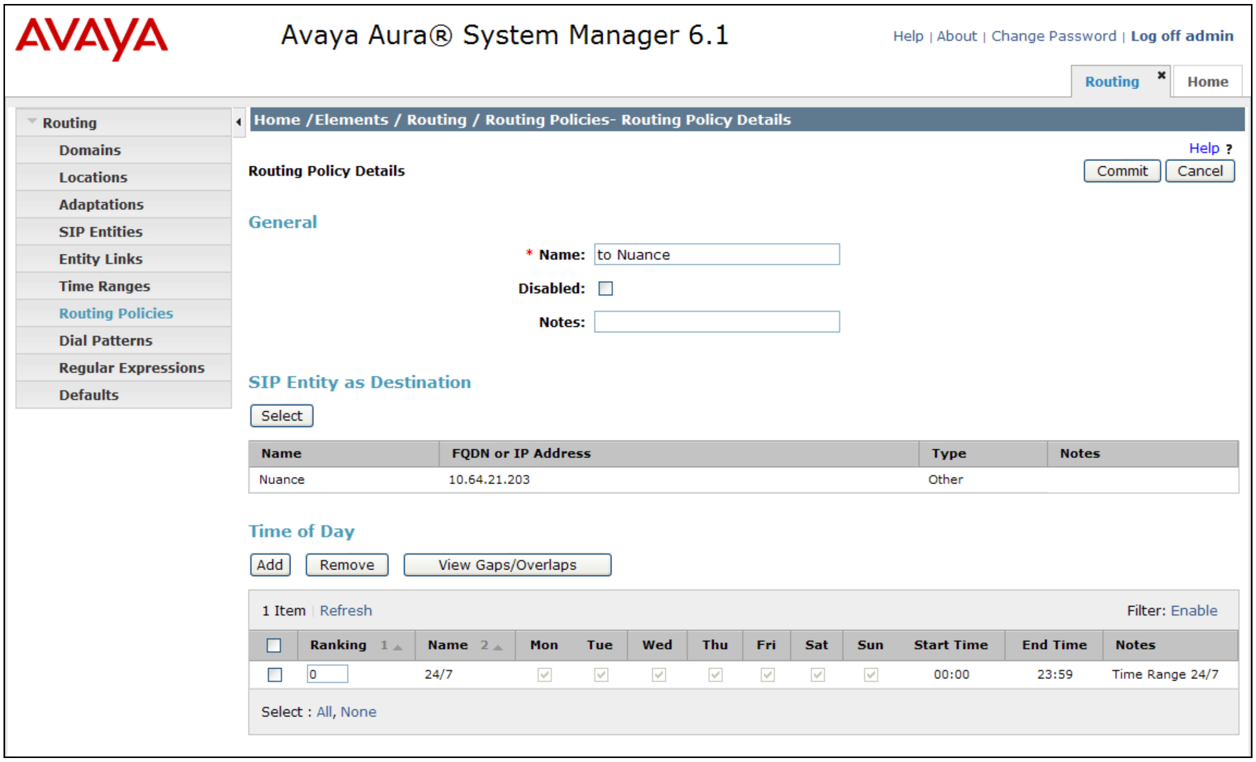
Under *General*:
Enter a descriptive name in **Name**.

Under *SIP Entity as Destination*:
Click **Select**, and then select the appropriate SIP entity to which this routing policy applies.

Defaults can be used for the remaining fields. Click **Commit** to save each Routing Policy definition. The following screen shows the Routing Policy for Communication Manager.



The following screen shows the Routing Policy for NVP.



6.6. Add Dial Patterns

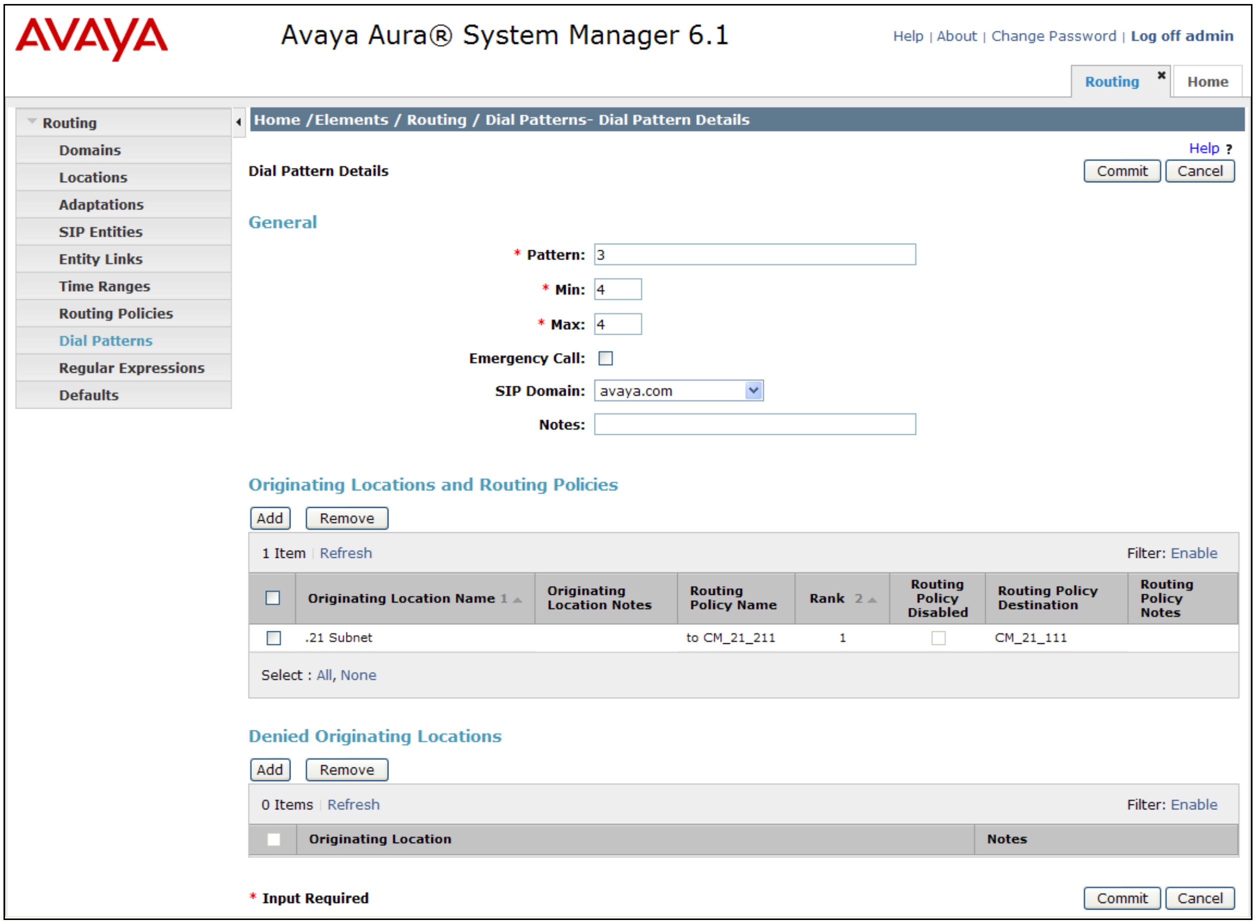
Dial patterns must be defined that will direct calls to the appropriate SIP Entity. In the sample configuration, 4-digit extensions beginning with "3" reside on Communication Manager, and extension "2222" is the NVP number. To add a dial pattern, navigate to **Routing → Dial Patterns** from the Home screen and click the **New** button (not shown) on the right. Fill in the following:

Under *General*:

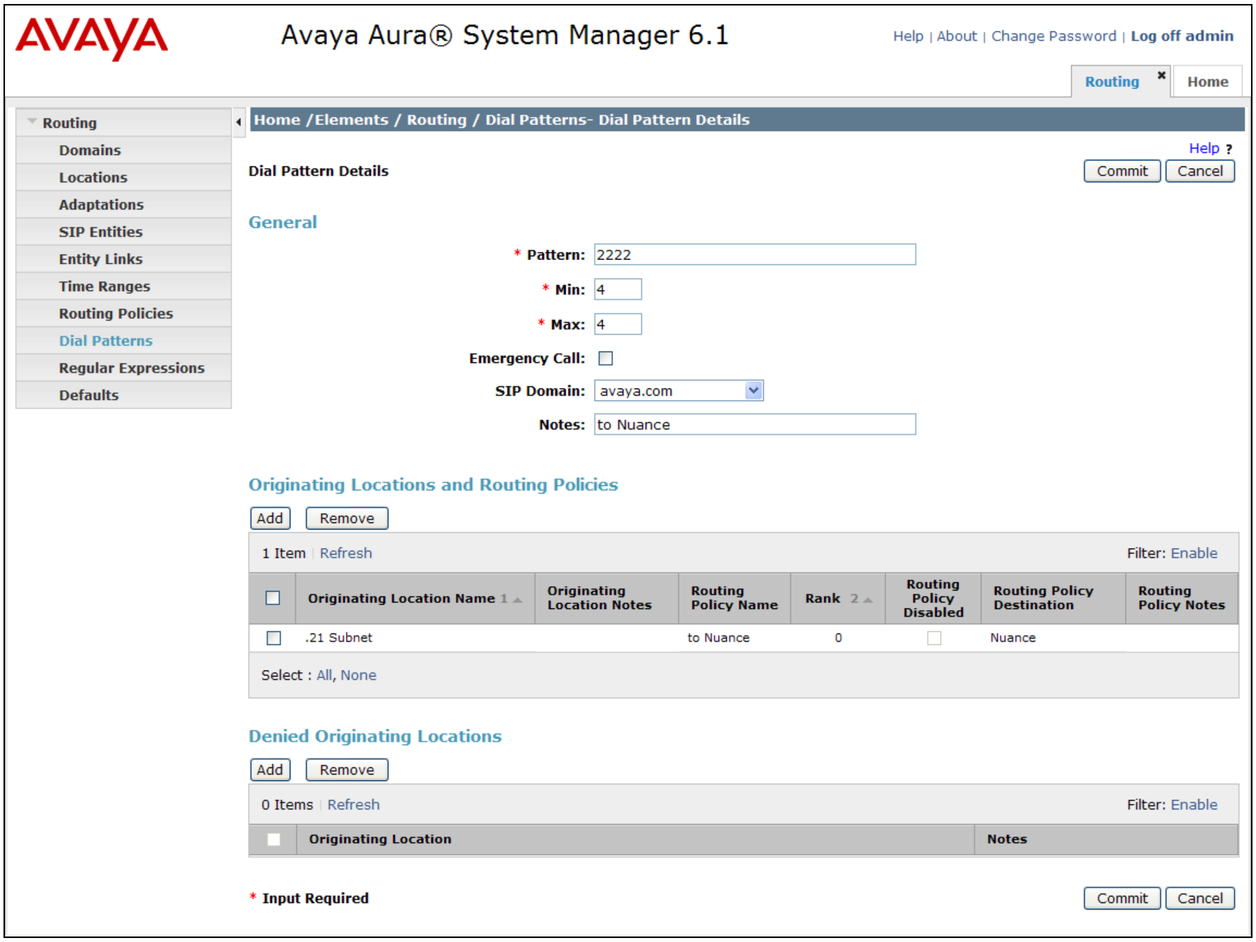
- **Pattern:** Dialed number or prefix.
- **Min** Minimum length of dialed number.
- **Max** Maximum length of dialed number.
- **SIP Domain** SIP domain of dial pattern.
- **Notes** Comment on purpose of dial pattern.

Under *Originating Locations and Routing Policies*:
Click **Add**, and then select the appropriate location and routing policy from the list.

Default values can be used for the remaining fields. The following screen shows the dial pattern definitions for local extensions on Communication Manager. Click **Commit** to save this dial pattern.


AVAYA
Avaya Aura® System Manager 6.1
Help | About | Change Password | Log off admin
Routing
Home
Routing
Domains
Locations
Adaptations
SIP Entities
Entity Links
Time Ranges
Routing Policies
Dial Patterns
Regular Expressions
Defaults
Home /Elements / Routing / Dial Patterns- Dial Pattern Details
Help ?
Dial Pattern Details
Commit
Cancel
General
* Pattern: 3
* Min: 4
* Max: 4
Emergency Call:
SIP Domain: avaya.com
Notes:
Originating Locations and Routing Policies
Add
Remove
1 Item | Refresh
Filter: Enable
Originating Location Name
Originating Location Notes
Routing Policy Name
Rank
Routing Policy Disabled
Routing Policy Destination
Routing Policy Notes
.21 Subnet
to CM_21_211
1
CM_21_111
Select : All, None
Denied Originating Locations
Add
Remove
0 Items | Refresh
Filter: Enable
Originating Location
Notes
* Input Required
Commit
Cancel

The following screen shows the dial pattern definition for the NVP number (2222).



6.7. Add Session Manager

To complete the configuration, adding Session Manager will provide the linkage between System Manager and Session Manager. From the Home screen, navigate to **Session Manager** → **Session ManagerAdministration** and then click the **New** button under **Session Manager Instances** on the right (not shown). Fill in the fields as described below and shown in the following screen:

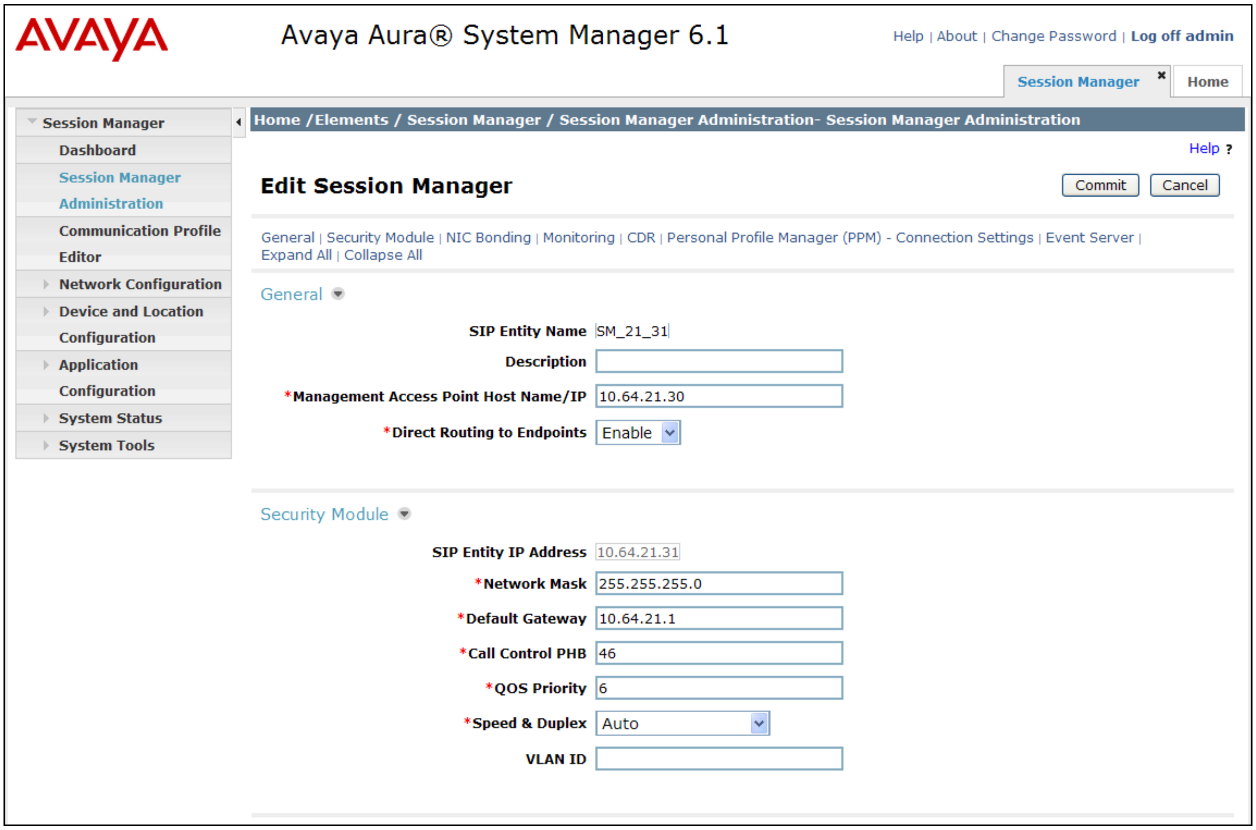
Under *Identity*:

- **SIP Entity Name:** Select the name of the SIP Entity added for Session Manager
- **Description:** Descriptive comment (optional)
- **Management Access Point Host Name/IP:** Enter the IP address of the Session Manager management interface.

Under *Security Module*:

- **Network Mask:** Enter the network mask corresponding to the IP address of Session Manager
- **Default Gateway**: Enter the IP address of the default gateway for Session Manager

Use default values for the remaining fields. Click **Save** to add this Session Manager.


Avaya Aura® System Manager 6.1
Help | About | Change Password | Log off admin
Session Manager
Home
Session Manager
Dashboard
Session Manager Administration
Communication Profile Editor
Network Configuration
Device and Location Configuration
Application Configuration
System Status
System Tools
Home /Elements / Session Manager / Session Manager Administration- Session Manager Administration
Help ?
Edit Session Manager
Commit
Cancel
General | Security Module | NIC Bonding | Monitoring | CDR | Personal Profile Manager (PPM) - Connection Settings | Event Server |
Expand All | Collapse All
General
SIP Entity Name SM_21_31
Description
*Management Access Point Host Name/IP 10.64.21.30
*Direct Routing to Endpoints Enable
Security Module
SIP Entity IP Address 10.64.21.31
*Network Mask 255.255.255.0
*Default Gateway 10.64.21.1
*Call Control PHB 46
*QOS Priority 6
*Speed & Duplex Auto
VLAN ID

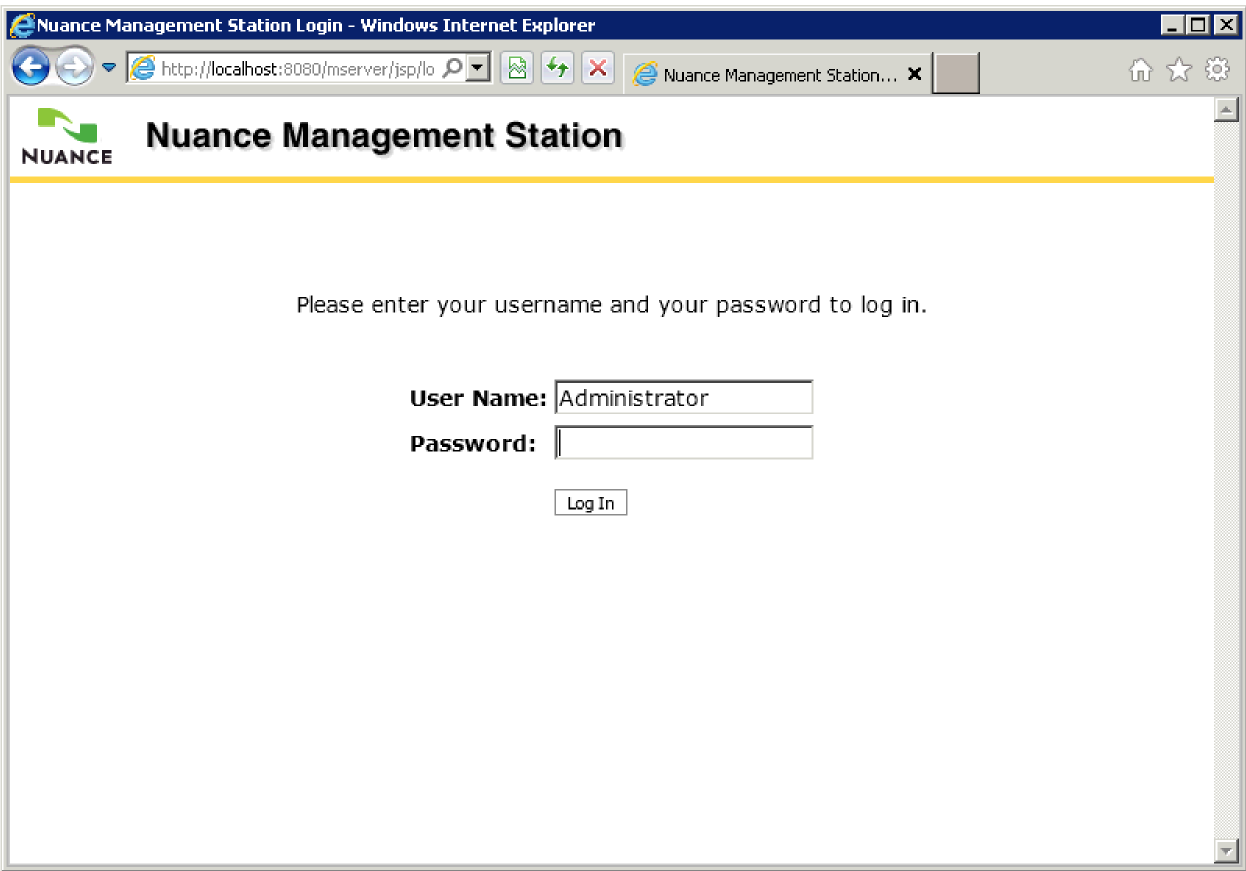
7. Configure Nuance Voice Platform

This section covers the procedure for configuring Nuance Voice Platform (NVP) and a custom application provided by Nuance. The procedure includes the following areas:

- Configure NVP via Nuance Management Station
- Configure the custom application provided by Nuance for compliance testing

7.1. Nuance Station Management

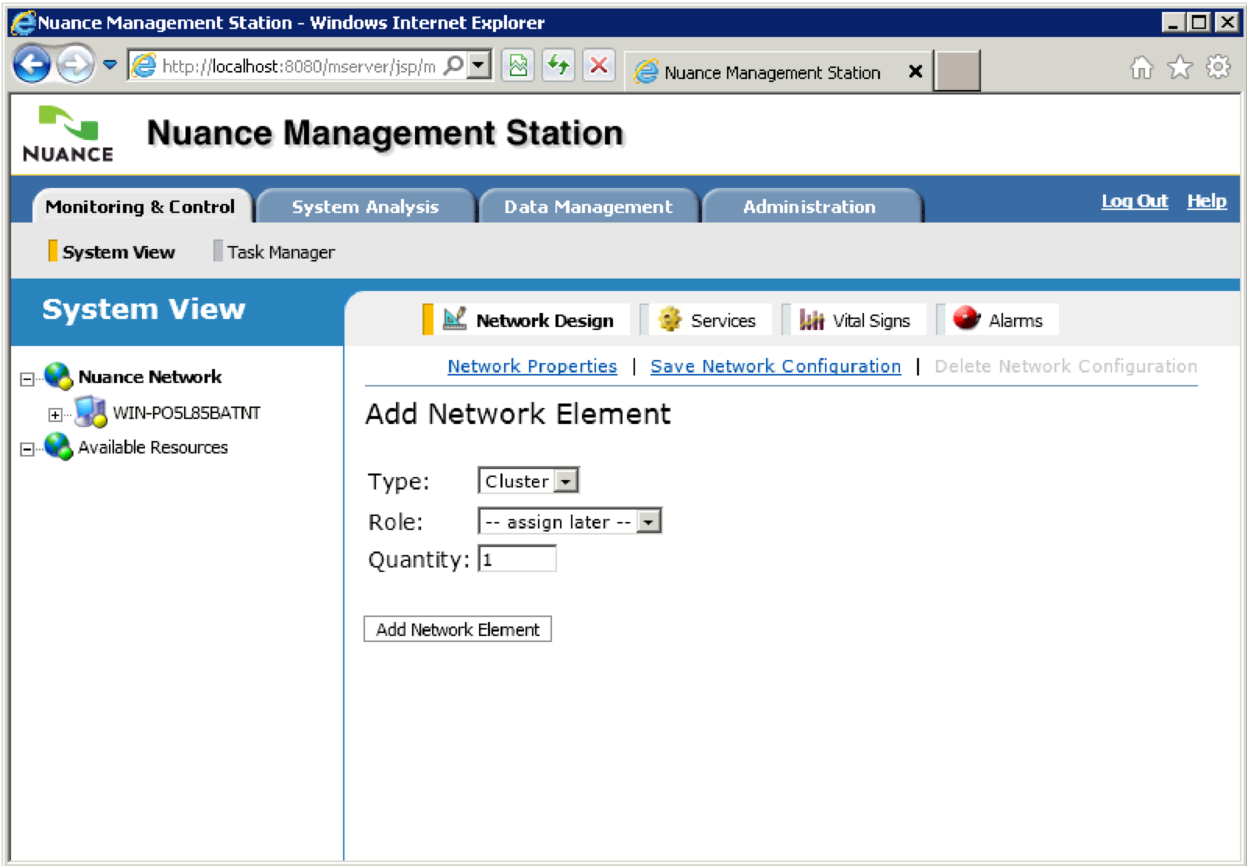
NVP is configured through **Nuance Management Station** which can be started by navigating to **Start → Programs → Nuance Voice Platform→ Management Station**. The initial screen is displayed below in a web browser.



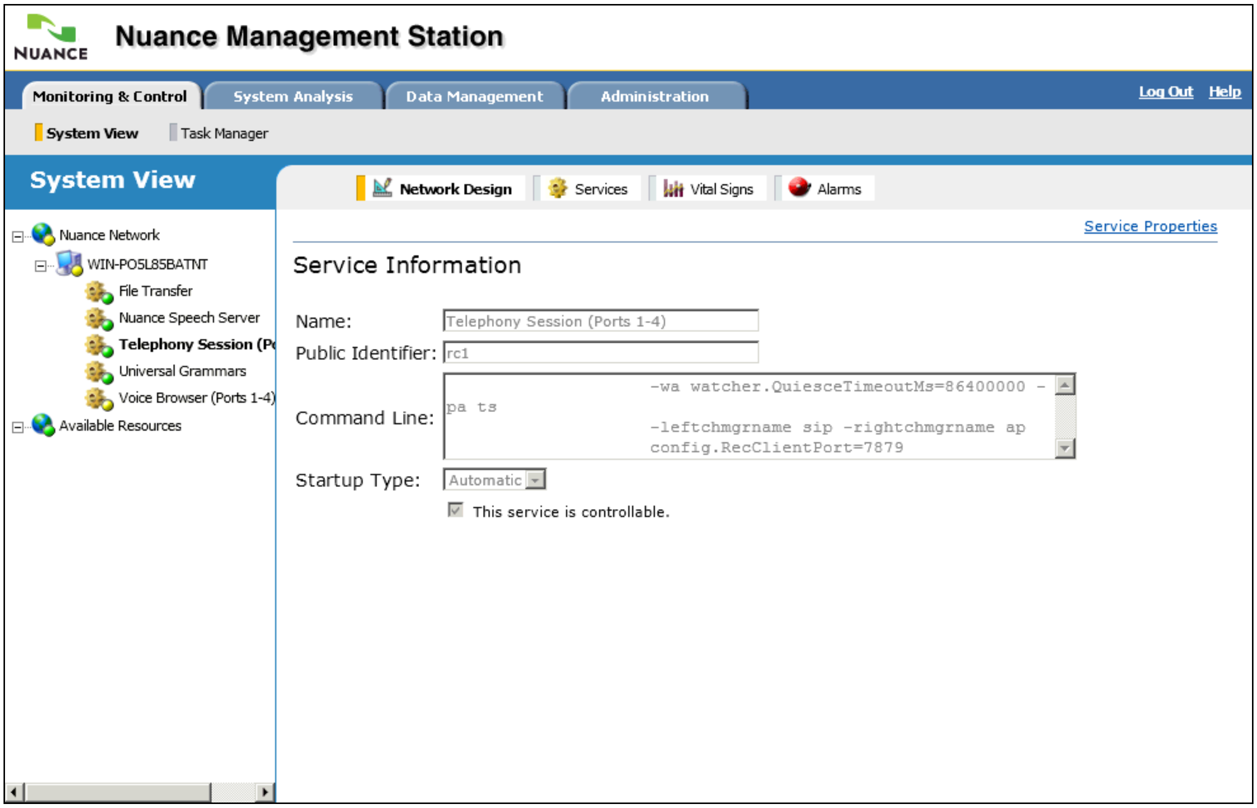
Alternatively, the Nuance Management Station can be accessed via a web browser from the local server by entering the following URL : http://localhost:8080/mserver.

It can also be accessed from another server that belongs to the same network via a web browser by entering the following URL: [http://*ip_of_NVP_Server*:8080/mserver](http://ip_of_NVP_Server:8080/mserver), where *ip_of_NVP_Server* must be change to the IP address of the NVP server.

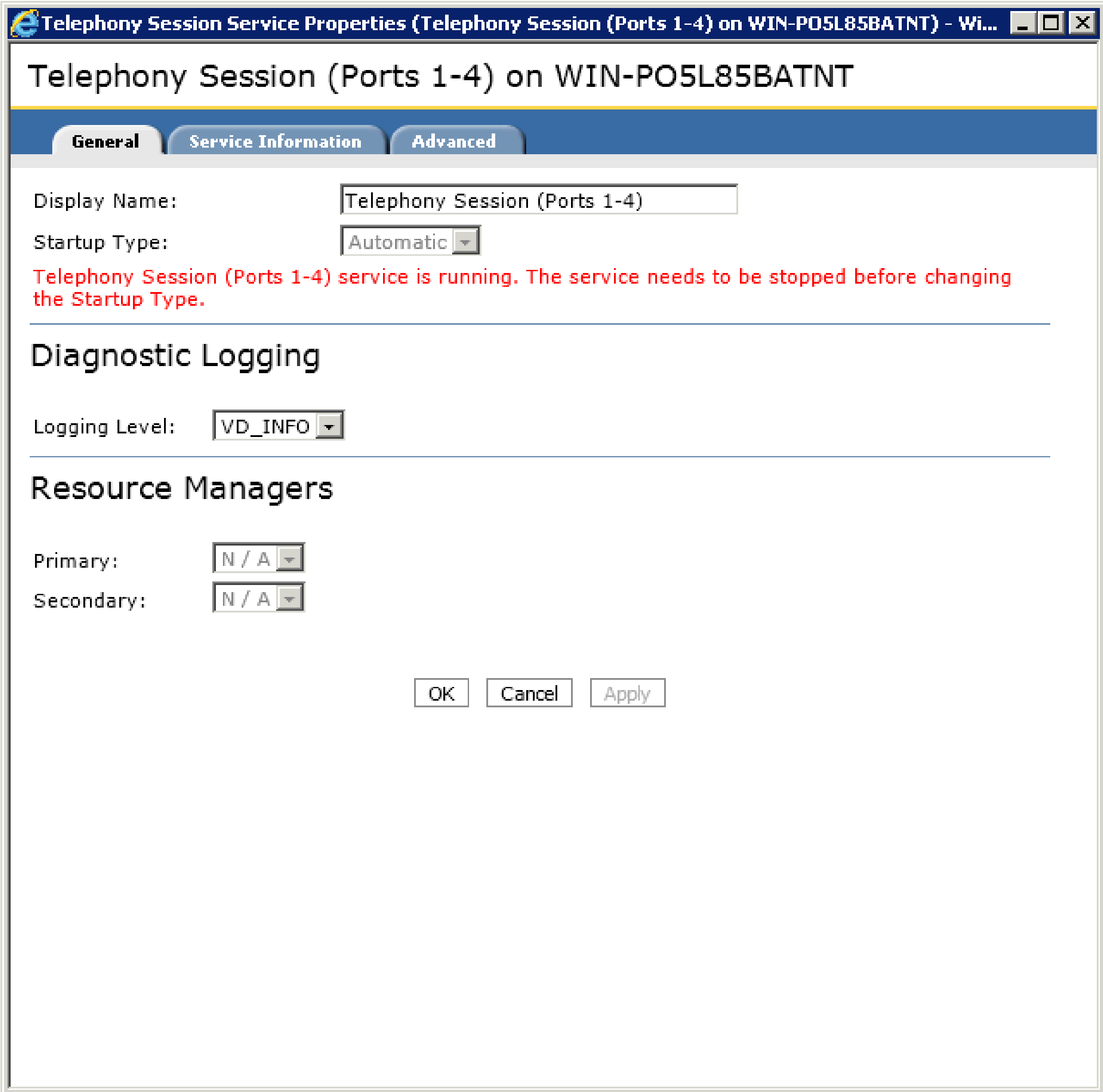
After logging in with appropriate credentials, the following screen is displayed:



Expand the server name on the left (e.g. *WIN-PO5L85BATNT*), and click the **Telephony Session** link. Click the **Service Properties** link towards the top right.



The following screen is displayed. Click the **Advanced** tab.



The following screen is displayed. Add or remove properties as required. The values shown below are the values used during compliance testing. Note the following:

- **audio.Provider** is set to *sip*.
- **audio.sip.Send503OnBusy** is set to *TRUE* to allow Session Manager to choose an alternate route when all the NVP ports are busy.
- **audio.sip.UserAgentPort** is set to *5060* to configure the SIP port that NVP listens on.
- **audio.sip.UserAgentURI** is set to *sip:nvp@%HOSTNAME%:5060* to specify the SIP URI for NVP.
- **ts.APRTPDtmfPayloadTyp**e is set to *127* to match the default value used by Communication Manager.
- **ts.APSIPGatewayList** is set to the IP address and port of the SIP signaling interface of Session Manager (i.e. *10.64.21.31:5060*).
- **ts.RTPBridge** is set to *TRUE*, but it is only required for bridge transfer call scenarios (no audio can be heard on bridge transfers unless **ts.RTPBridge** is set to TRUE).

7.2. Custom Application

A custom application (nvp4_certification.vxml) was provided by Nuance to compliance test the NVP solution. The application was stored at the following path on the NVP server:

C:\Program Files (x86)\Nuance\Voice Platform\Speech Server\session\nvp4_certification.vxml

The complete contents of the file are included in the Appendix. The following parameters within the file were modified with the values shown below for compliance testing:

```
<var name="gatewayaddress"          expr="'10.64.21.31'"/>
<var name="answertransferext"       expr="'3001'"/>
<var name="busytransferext"         expr="'3002'"/>
<var name="noanswertransferext"     expr="'3001'"/>
<var name="invalidtransferext"      expr="'5555'"/>
```

8. Verification Steps

This section provides the verification steps that may be performed to verify that NVP is operating properly with Session Manager and Communication Manager using SIP integration.

1. Place a call to NVP. Verify that the NVP greeting is heard.
2. Speak a menu selection or enter the menu selection via DTMF to transfer the call. Verify NVP transfers the call to the proper destination.

9. Conclusion

These Application Notes describe the configuration steps required to integrate Nuance Voice Platform with Avaya Aura® Session Manager and Avaya Aura® Communication Manager using SIP integration. All feature and serviceability test cases were completed successfully.

10. Additional References

This section references the Avaya documentation relevant to these Application Notes. The following Avaya product documentation is available at http://support.avaya.com.

[1] *Administering Avaya Aura® Communication Manager*, May 2009, Release 5.2, Issue 5.0, Document Number 03-300509.

[2] *Administering Avaya Aura® Session Manager*, May 2011, Issue 3, Release 6.1, Document Number 03-603324.

Nuance product documentation is available by contacting Nuance at http://www.nuance.com/support/.

11. Appendix – Nuance Custom Application

The complete contents of the custom application file (*nvp4_certification.vxml*) provided by Nuance for compliance testing is shown below:

```
<?xml version="1.0"?>


<!DOCTYPE vxml PUBLIC "-//Nuance/DTD VoiceXML 2.0//EN"
"http://voicexml.nuance.com/dtd/nuancevoicexml-2-0.dtd">

<vxml version="2.0">
<meta http-equiv="Cache-Control" content="no-cache"/>

    <property name="audiomaxage"       value="0"/>
    <property name="audiomaxstale"     value="0"/>
    <property name="grammarmaxage"     value="0"/>
    <property name="grammarmaxstale"   value="0"/>
    <property name="documentmaxage"    value="0"/>
    <property name="documentmaxstale"  value="0"/>

    <!-- This is needed for hotword to work with multi-word sentences
    -->
    <property name="nuance.maxHotwordDuration" value="5s"/>

    <var name="telPort"/>
    <var name="apType"/>
    <var name="callingUri"/>
    <var name="calledUri"/>
    <var name="testNumber"/>
    <var name="testName"/>
    <var name="callerType" expr="''"/>
    <var name="testResults" expr="''"/>


    <!-- TODO: Edit the following to match the PBX or gateway setup -->

    <var name="gatewayaddress"          expr="'10.64.21.31'"/>
    <var name="answertransferext"       expr="'3001'"/>
    <var name="busytransferext"         expr="'3002'"/>
    <var name="noanswertransferext"     expr="'3001'"/>
    <var name="invalidtransferext"      expr="'5555'"/>
    <var name="postdialext"             expr="'2472'"/>
    <var name="transferaai"             expr="'aaiTest'"/>

    <!-- End of PBX/gateway specific changes -->


    <var name="answertransferdest"      expr="'sip:' + answertransferext + '@' +
gatewayaddress"/>
    <var name="busytransferdest"        expr="'sip:' + busytransferext + '@' +
gatewayaddress"/>
    <var name="noanswertransferdest"    expr="'sip:' + noanswertransferext + '@' +
gatewayaddress"/>
    <var name="invalidtransferdest"     expr="'sip:' + invalidtransferext + '@' +
gatewayaddress"/>

    <var name="heSaid"                  expr="'nothing'"/>
    <var name="attempts"                expr="0"/>
    <var name="maxAttempts"             expr="10"/>
```

```
<script>
    <![CDATA[
    function splitUri(uri)
    {
        var userPart, hostPart, portPart;
        var userStart = uri.indexOf(':');
        if (userStart > 0)
        {
            userStart++; // skip :
            var userEnd = uri.indexOf('@', userStart + 1);
            if (userEnd > 0)
            {
                userPart = uri.substring(userStart, userEnd);
                userEnd++; // skip @
                var hostEnd = uri.indexOf(':', userEnd + 1);
                if (hostEnd > 0)
                {
                    hostPart = uri.substring(userEnd, hostEnd);
                    hostEnd++; // skip :
                    portPart = uri.substring(hostEnd);
                }
                else
                {
                    hostPart = uri.substring(userEnd);
                    portPart = "";
                }
            }
        }
        return {userPart : userPart, hostPart : hostPart, portPart : portPart};
    }

    function getTestName(testNumber)
    {
        if (testNumber == "01")
        {
            testName = "hanguptest";
        }
        else if (testNumber == "02")
        {
            testName = "record_test";
        }
        else if (testNumber == "03")
        {
            testName = "blindtransfertest";
        }
        else if (testNumber == "04")
        {
            testName = "blindtransferextest";
        }
        else if (testNumber == "05")
        {
            testName = "conditionaltransfertest";
        }
        else if (testNumber == "06")
        {
            testName = "bridgetransfertest";
        }
        else if (testNumber == "07")
        {
            testName = "consultationtransfertest";
        }
```

```
else if (testNumber == "08")
{
    testName = "fedtransfertest";
}
else if (testNumber == "09")
{
    testName = "hotwordtransfertest";
}
else if (testNumber == "0A")
{
    testName = "dtmfatest";
}
else if (testNumber == "10")
{
    testName = "consultfedtransfertest";
}
else if (testNumber == "11")
{
    testName = "failedblindtransfertest";
}
else if (testNumber == "12")
{
    testName = "failedconditionaltransfertest";
}
else if (testNumber == "13")
{
    testName = "busyblindtransfertest";
}
else if (testNumber == "14")
{
    testName = "busyconditionaltransfertest";
}
else if (testNumber == "15")
{
    testName = "busyconsultationtransfertest";
}
else if (testNumber == "16")
{
    testName = "busybridgetransfertest";
}
else if (testNumber == "17")
{
    testName = "postdialtest";
}
else if (testNumber == "18")
{
    testName = "playdtmftest";
}
else if (testNumber == "19")
{
    testName = "hotwordtest";
}
else if (testNumber == "20")
{
    testName = "amdetectiontest";
}
else if (testNumber == "22")
{
    testName = "noanswerbridgetransfertest";
}
else if (testNumber == "27")
{
```

```
                testName = "recordhotwordtest";
            }
            else
            {
                testName = "";
            }
        }

        function prepareTestVars()
        {
            if (telPort) {
                var apParam = "ap=";
                var start = telPort.indexOf(apParam);
                start += apParam.length;
                var end = telPort.indexOf(',', start + 1);
                apType = telPort.substring(start, end);
            }

            if (apType == "nms")
            {

            }
            else
            {


            }
        }

        // Init code
        testResults += "session.connection=";
        var x;
        for (x in session.connection)
        {
            testResults += x;
            testResults += ", ";
        }
        testResults += ", ";
        testResults += "session.connection.protocol=" + session.connection.protocol +
", ";
        testResults += "session.connection.protocol=";
        for (x in session.connection.protocol)
        {
            testResults += x;
            testResults += ", ";
        }
        testResults += ", ";

        testResults += "session.connection.protocol.sip=" +
session.connection.protocol.sip + "; ";
        testResults += "session.connection.protocol.name=" +
session.connection.protocol.name + "; ";
        testResults += "session.connection.protocol.version=" +
session.connection.protocol.version + "; ";

        testResults += "session.connection.ctiport=" + session.connection.ctiport + ",
";
        testResults += "session.connection.device=" + session.connection.device + ",
";
        testResults += "session.connection.redirect=" + session.connection.redirect +
", ";
        testResults += "session.connection.local=" + session.connection.local + ", ";
```

```
        testResults += "session.connection.remote=" + session.connection.remote + ", 
";
        testResults += "session.connection.aai=" + session.connection.aai + ", ";
        testResults += "session.connection.sessionid=" + session.connection.sessionid 
+ ", ";
        testResults += "session.connection.originator=" + 
session.connection.originator + ", ";
        testResults += "session.connection.telephonyport=" + 
session.connection.telephonyport + ", ";

        telPort = session.connection.telephonyport;
        callingUri = splitUri(session.connection.remote.uri);
        calledUri = splitUri(session.connection.local.uri);

        ]]>
    </script>

    <catch event="connection.disconnect">
        <assign name="testResults" expr="testResults + ', disconnect_event=' + 
_event"/>
        <goto next="#exitform"/>
    </catch>

    <catch event="nuance.interrupt.device">
        <assign name="testResults" expr="testResults + 'device_event=' + _event + ', 
'"/>
    </catch>

    <catch event="nuance.interrupt.aai">
        <assign name="testResults" expr="testResults + 'aai_event=' + _event + ', '"/>
    </catch>

    <form id="main_menu">

        <!-- Go to a specific test -->
        <block>
            <log>AAA entering main_menu</log>

            <assign name="attempts" expr="attempts + 1"/>
            <if cond="attempts > maxAttempts">
                Goodbye.
                <disconnect/>
            </if>

            <script>
            <![CDATA[
                if (calledUri.userPart.substring(0, 1) == "1")
                   testNumber = calledUri.userPart.substring(2, 4);
                else
                   testNumber = "00";
                getTestName(testNumber);
            ]]>
            </script>

            <if cond="testName == ''">
                <goto next="#choose_test"/>
            <else/>
                <goto next="#runTest"/>
            </if>

        </block>
    </form>
```

```
    <form id="runTest">
        <block>
            <script>
            <![CDATA[
                prepareTestVars();
            ]]>
            </script>

            <assign name="testResults" expr="testResults + ', test=' + testName + ',
calledUri.userPart=' + calledUri.userPart + ', '"/>
            <goto expr="'#' + testName"/>
        </block>
    </form>


    <form id="choose_test">
        <property name="inputmodes"        value="voice dtmf"/>
        <property name="interdigittimeout" value="1500ms"/>
        <property name="termtimeout"       value="1500ms"/>
        <property name="termchar"          value="#"/>
        <link event="dtmf">
            <grammar mode="dtmf" root="dtmf_main">
                <rule id="dtmf_main" scope="public">
                    <one-of>
                        <item>
                            <item>1</item>
                            <tag>RESULT="01"</tag>
                        </item>
                        <item>
                            <item>2</item>
                            <tag>RESULT="02"</tag>
                        </item>
                        <item>
                            <item>3</item>
                            <tag>RESULT="03"</tag>
                        </item>
                        <item>
                            <item>4</item>
                            <tag>RESULT="04"</tag>
                        </item>
                        <item>
                            <item>5</item>
                            <tag>RESULT="05"</tag>
                        </item>
                        <item>
                            <item>6</item>
                            <tag>RESULT="06"</tag>
                        </item>
                        <item>
                            <item>7</item>
                            <tag>RESULT="07"</tag>
                        </item>
                        <item>
                            <item>8</item>
                            <tag>RESULT="08"</tag>
                        </item>
                        <item>
                            <item>9</item>
                            <tag>RESULT="09"</tag>
                        </item>
                        <item>
```

```
        <item>A</item>
        <tag>RESULT="0A"</tag>
    </item>
    <item>
        <item>1 0</item>
        <tag>RESULT="10"</tag>
    </item>
    <item>
        <item>1 1</item>
        <tag>RESULT="11"</tag>
    </item>
    <item>
        <item>1 2</item>
        <tag>RESULT="12"</tag>
    </item>
    <item>
        <item>1 3</item>
        <tag>RESULT="13"</tag>
    </item>
    <item>
        <item>1 4</item>
        <tag>RESULT="14"</tag>
    </item>
    <item>
        <item>1 5</item>
        <tag>RESULT="15"</tag>
    </item>
    <item>
        <item>1 6</item>
        <tag>RESULT="16"</tag>
    </item>
    <item>
        <item>1 7</item>
        <tag>RESULT="17"</tag>
    </item>
    <item>
        <item>1 8</item>
        <tag>RESULT="18"</tag>
    </item>
    <item>
        <item>1 9</item>
        <tag>RESULT="19"</tag>
    </item>
    <item>
        <item>2 0</item>
        <tag>RESULT="20"</tag>
    </item>
    <item>
        <item>2 1</item>
        <tag>RESULT="21"</tag>
    </item>
    <item>
        <item>2 2</item>
        <tag>RESULT="22"</tag>
    </item>
    <item>
        <item>2 3</item>
        <tag>RESULT="23"</tag>
    </item>
    <item>
        <item>2 4</item>
        <tag>RESULT="24"</tag>
```

```
                </item>
                <item>
                    <item>2 5</item>
                    <tag>RESULT="25"</tag>
                </item>
                <item>
                    <item>2 6</item>
                    <tag>RESULT="26"</tag>
                </item>
                <item>
                    <item>2 7</item>
                    <tag>RESULT="27"</tag>
                </item>

            </one-of>
        </rule>
    </grammar>
</link>

<catch event="dtmf">
    <script>
    <![CDATA[
        testNumber = application.lastresult$[0].interpretation.RESULT;
        getTestName(testNumber);
    ]]>
    </script>
    <goto next="#runTest"/>
</catch>

<field name="menu">

    <grammar mode="voice" root="voice_main">
        <rule id="voice_main" scope="public">
            <one-of>
                <item>Abby Armstrong</item>
                <item>Adrian Polack</item>
                <item>
                    <item>hangup</item>
                    <tag>RESULT="01"</tag>
                </item>
                <item>
                    <item>record</item>
                    <tag>RESULT="02"</tag>
                </item>
                <item>
                    <item>blind transfer</item>
                    <tag>RESULT="03"</tag>
                </item>
                <item>
                    <item>blind transfer ex</item>
                    <tag>RESULT="04"</tag>
                </item>
                <item>
                    <item>conditional transfer</item>
                    <tag>RESULT="05"</tag>
                </item>
                <item>
                    <item>bridge transfer</item>
                    <tag>RESULT="06"</tag>
                </item>
                <item>
                    <item>consultation transfer</item>
```

```
                <tag>RESULT="07"</tag>
            </item>
            <item>
                <item>far end dialog transfer</item>
                <tag>RESULT="08"</tag>
            </item>
            <item>
                <item>hot word transfer</item>
                <tag>RESULT="09"</tag>
            </item>
            <item>
                <item>consultation far end dialog transfer</item>
                <tag>RESULT="10"</tag>
            </item>
            <item>
                <item>failed blind transfer</item>
                <tag>RESULT="11"</tag>
            </item>
            <item>
                <item>failed conditional transfer</item>
                <tag>RESULT="12"</tag>
            </item>
            <item>
                <item>busy blind transfer</item>
                <tag>RESULT="13"</tag>
            </item>
            <item>
                <item>busy conditional transfer</item>
                <tag>RESULT="14"</tag>
            </item>
            <item>
                <item>busy consultation transfer</item>
                <tag>RESULT="15"</tag>
            </item>
            <item>
                <item>busy bridge transfer</item>
                <tag>RESULT="16"</tag>
            </item>
            <item>
                <item>post dial</item>
                <tag>RESULT="17"</tag>
            </item>
            <item>
                <item>play dtmf</item>
                <tag>RESULT="18"</tag>
            </item>
            <item>
                <item>hotword</item>
                <tag>RESULT="19"</tag>
            </item>
            <item>
                <item>answering machine detection</item>
                <tag>RESULT="20"</tag>
            </item>
            <item>
                <item>no answer bridge transfer</item>
                <tag>RESULT="22"</tag>
            </item>

        </one-of>
    </rule>
</grammar>
```

```
            <prompt bargein="true" timeout="10s">
                Main menu hangup, record, blind transfer, blind transfer ex,
conditional transfer, consultation transfer, bridge transfer, far end dialog transfer,
hot word transfer, consultation far end dialog transfer, failed blind transfer, failed
conditional transfer, busy blind transfer, busy conditional transfer, busy
consultation transfer, busy bridge transfer, post dial, hotword, answering machine
detection, no answer bridge transfer
            </prompt>
            <filled>
                <script>
                <![CDATA[
                    testNumber = application.lastresult$[0].interpretation.RESULT;
                    getTestName(testNumber);
                ]]>
                </script>
                <goto next="#runTest"/>
            </filled>
        </field>
    </form>

    <form id="hanguptest">
        <block>
            Goodbye
            <audio src="pause:3000"/>
        </block>
        <block>
           <disconnect/>
        </block>
    </form>

    <form id="record_test">
        <record name="msg" beep="true" type="audio/x-wav" maxtime="10000ms"
modal="true" finalsilence="5000ms" dtmfterm="true">
            <prompt>
                Record a message after the beep.
            </prompt>
            <noinput>
                I didn't hear anything, please try again.
            </noinput>
            <filled>
                <prompt>You said <audio expr="msg"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </record>
    </form>

    <form id="blindtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="blind" destexpr="answertransferdest">
            <prompt>Doing blind transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
```

```
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="blindtransferextest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
            <assign name="testResults" expr="testResults + ', aai = ' + transferaai"/>
        </block>
        <transfer name="call" type="blind" destexpr="answertransferdest"
aaiexpr="transferaai">
            <prompt>Doing blind transfer with a. a. i.</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="conditionaltransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="conditional" destexpr="answertransferdest">
            <prompt>Doing conditional transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="bridgetransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="answertransferdest"
localuriexpr="session.connection.local.uri">
            <prompt>Doing bridge transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
```

```
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="consultationtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="consultation" destexpr="answertransferdest"
localuriexpr="session.connection.local.uri">
            <prompt>Doing a consultation transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="fedtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="answertransferdest"
localuriexpr="session.connection.local.uri" farenddialog="#confirm_transfer">
            <prompt>Doing a far end dialog transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="consultfedtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="consultation" destexpr="answertransferdest"
localuriexpr="session.connection.local.uri" farenddialog="#confirm_transfer">
            <prompt>Doing a consultation far end dialog transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
```

```
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="confirm_transfer">
        <field name="confirmation" type="boolean">
            <prompt>Do you accept the call?</prompt>
            <filled>
                <if cond="confirmation == 'yes'">
                    <assign name="confirmation" expr="true"/>
                </if>
                <if cond="confirmation == 'true'">
                    <assign name="confirmation" expr="true"/>
                </if>
                <if cond="confirmation == true">
                    <assign name="heSaid" expr="'yeah'"/>
                    Connecting
                <else/>
                    <assign name="heSaid" expr="'neah'"/>
                    Refusing
                </if>
                <return namelist="confirmation"/>
            </filled>
        </field>
    </form>

    <form id="hotwordtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="answertransferdest">
            <grammar type="application/srgs+xml" root="Hotword">
                <rule id="Hotword">
                    <one-of>
                        <item>please stop this transfer</item>
                    </one-of>
                </rule>
            </grammar>
            <grammar type="application/srgs+xml" root="HotDtmf" mode="dtmf">
                <rule id="HotDtmf">
                    <one-of>
                        <item>1 2</item>
                    </one-of>
                </rule>
            </grammar>
            <prompt>Doing a hot word transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>
```

```
    <form id="dtmfatest">
        <block>
            DTMF A entered, goodbye
            <audio src="pause:1000"/>
            <goto next="#main_menu"/>
        </block>
    </form>

    <form id="failedblindtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', invalidtransferdest = '
+ invalidtransferdest"/>
        </block>
        <transfer name="call" type="blind" destexpr="invalidtransferdest">
            <prompt>Doing failed blind transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="failedconditionaltransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', invalidtransferdest = '
+ invalidtransferdest"/>
        </block>
        <transfer name="call" type="conditional" destexpr="busytransferdest">
            <prompt>Doing failed conditional transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="busyblindtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', busytransferdest = ' +
busytransferdest"/>
        </block>
        <transfer name="call" type="blind" destexpr="busytransferdest">
            <prompt>Doing busy blind transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
```

```
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="busyconditionaltransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', busytransferdest = ' +
busytransferdest"/>
        </block>
        <transfer name="call" type="conditional" destexpr="busytransferdest">
            <prompt>Doing busy conditional transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="busyconsultationtransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', busytransferdest = ' +
busytransferdest"/>
        </block>
        <transfer name="call" type="consultation" destexpr="busytransferdest">
            <prompt>Doing busy consultation transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="busybridgetransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', busytransferdest = ' +
busytransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="busytransferdest">
            <prompt>Doing busy bridge transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
```

```
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="noanswerbridgetransfertest">
        <block>
            <assign name="testResults" expr="testResults + ', noanswertransferdest = '
+ noanswertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="noanswertransferdest"
connecttimeout="4s">
            <prompt>Doing no answer bridge transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="postdialtest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="answertransferdest"
connecttimeout="30s" farenddialog="#postdialfed">
            <prompt>Doing a post dial</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="postdialfed">
        <block>
            <audio src="pause:2000"/>
            <audio expr="'dtmf:' + postdialext"/>
            <var name="confirmation" expr="true"/>
            <return namelist="confirmation"/>
        </block>
    </form>

    <form id="playdtmftest">
        <block>
            Playing DTMF
            <audio src="dtmf:1"/>
            <audio src="pause:100"/>
            <audio src="dtmf:2"/>
```

```
            <audio src="pause:100"/>
            <audio src="dtmf:3"/>
            <audio src="pause:100"/>
            <audio src="dtmf:4"/>
            <audio src="pause:100"/>
            <goto next="#main_menu"/>
        </block>
    </form>

    <form id="hotwordtest">
        <field name="hotwordfield">
            <grammar type="application/srgs+xml" root="HotwordDtmf" mode="dtmf">
                <rule id="HotwordDtmf">
                    <one-of>
                        <item>1 2 3 4</item>
                    </one-of>
                </rule>
            </grammar>
            <grammar type="application/srgs+xml" root="HotwordVoice" mode="voice">
                <rule id="HotwordVoice">
                    <one-of>
                        <item>take me back</item>
                    </one-of>
                </rule>
            </grammar>
            <prompt bargein="true" bargeintype="hotword">
                This is a hotword test, enter dtmf 1 2 3 4 or say "take me back". This
long prompt will continue to play while entering the DTMF and if they aren't
recognized.
            </prompt>
            <filled>
                <prompt>Got it</prompt>
                <goto next="#main_menu"/>
            </filled>
        </field>
    </form>

    <form id="amdetectiontest">
        <block>
            <assign name="testResults" expr="testResults + ', answertransferdest = ' +
answertransferdest"/>
        </block>
        <transfer name="call" type="bridge" destexpr="answertransferdest"
farenddialog="#amconfirm">
            <prompt>Doing an A M detection transfer</prompt>
            <filled>
                <assign name="testResults" expr="testResults + ', filled=' + call"/>
                <prompt>filled value is <value expr="call"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </transfer>
        <catch event="error">
            <assign name="testResults" expr="testResults + ', event=' + _event"/>
            <prompt>event value is <value expr="_event"/></prompt>
            <goto next="#main_menu"/>
        </catch>
    </form>

    <form id="amconfirmtest">
        <subdialog src="#amconfirm"/>
    </form>
```

```
    <form id="amconfirm">
        <var name="confirmation" expr="false"/>
        <record name="AnsweringMachineDetection" maxtime="15s" finalsilence="2000ms"
beep="false" type="audio/x-wav" dtmfterm="false">
            <property name="sensitivity" value=".5"/>
            <property name="maxspeechtimeout" value="15s"/>
            <property name="timeout" value="15s"/>
            <filled>
                <if cond="AnsweringMachineDetection$.duration >= '4000'">
                    <assign name="confirmation" expr="false"/>
                <else/>
                    <assign name="confirmation" expr="true"/>
                </if>
                <if cond="confirmation == true">
                    Got a human, connecting.
                <else/>
                    Got an answering machine, refusing.
                </if>
                <return namelist="confirmation"/>
            </filled>
        </record>
    </form>

    <form id="recordhotwordtest">
        <record name="recordhotwordmsg" beep="true" type="audio/x-wav"
maxtime="20000ms" modal="true" finalsilence="15000ms">
            <grammar type="application/srgs+xml" root="recordhotwordgrammardtmf"
mode="dtmf">
                <rule id="recordhotwordgrammardtmf">
                    <one-of>
                        <item>1 2 3 4</item>
                    </one-of>
                </rule>
            </grammar>
            <grammar type="application/srgs+xml" root="recordhotwordgrammarvoice"
mode="voice">
                <rule id="recordhotwordgrammarvoice">
                    <one-of>
                        <item>goodbye</item>
                    </one-of>
                </rule>
            </grammar>
            <prompt>
                Record a message after the beep.
            </prompt>
            <noinput>
                I didn't hear anything, please try again.
            </noinput>
            <filled>
                <prompt>You said <audio expr="recordhotwordmsg"/></prompt>
                <goto next="#main_menu"/>
            </filled>
        </record>
    </form>

    <form id="exitform">
        <block>
            <log label="trace:?level=MINOR_ALARM" expr="testResults"/>
        </block>
    </form>

</vxml>
```

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at devconnect@avaya.com.